

初学者学习 FPGA，必定会被它的各种仿真弄的晕头转向。比如，前仿真、后仿真、功能仿真、时序仿真、行为级仿真、RTL 级仿真、综合后仿真、门级仿真、布局布线后仿真等。

Quartus 和 Modelsim 软件的仿真形式

Quartus II 有两种仿真形式：1、功能仿真；2、时序仿真。

Quartus II 调用 Modelsim 的两种仿真形式为：1、RTL 级仿真；2、Gate-level 仿真。

以下内容均经过资料查证，详细如下：

理解方法一

当用 quartus 进行仿真时，分为功能仿真（al）和时序仿真（Timing）；

当用 Modelsim-Altera 时，分为功能仿真（RTL）、综合后仿真（post-synthesis）和布局布线仿真（Gate-level）。其中，功能仿真又称为前仿真，布局布线仿真又称为后仿真。

Modelsim-Altera 仿真一般分为功能仿真，前仿真（综合后仿真）与后仿真（时序仿真或布局布线后仿真）。

根据设计需要，编写完代码（Verilog hdl, Vhdl, system Verilog）后，首先进行功能仿真，验证所写代码是否能完成设计功能；前仿真又称为综合后仿真，即在 QuartusII 完成综合后，验证设计的功能；后仿真又称为时序仿真或布局布线后仿真，是加入延时后的仿真。对于编译时间较短的小规模设计，一般只进行功能仿真与后仿真。

理解方法二

modelsim 是专门进行仿真的软件，可以分别进行前仿真和后仿真。前仿真也称为功能仿真，主旨在于验证电路的功能是否符合设计要求，其特点是不考虑电路门延迟与线延迟，主要是验证电路与理想情况是否一致。可综合 FPGA 代码是用 RTL 级代码语言描述的，其输入为 RTL 级代码与 testbench。后仿真也称为时序仿真或者布局布线后仿真，是指电路已经映射到特定的工艺环境以后，综合考虑电路的路径延迟与门延迟的影响，验证电路能否在一定时序条件下满足设计构想的过程，是否存在时序违规。其输入文件为从布局布线结果抽象出来的门级网表、testbench 和扩展为 sdo 或 sdf 的标准时延文件。sdo、sdf 的标准时延文件不仅包含门延迟，还包括实际布线延迟，能较好地反映芯片的实际工作情况。一般来说后仿真是必选的，检查设计时序与实际的 FPGA 运行情况是否一致，确保设计的可靠性和稳定性。

理解方法三

前仿真和后仿真的区别：前仿真就是指综合前的仿真，也就是行为级的仿真，如你在 Modelsim 直接写代码的仿真。后仿真指的是综合后的仿真，也就是功能仿真。比如你在 Modelsim 中用 VHDL 写了个计数器，行为级得仿真通过了，你把它加到 quartus 中或者其他的综合工具进行综合，综合完后生成功能网表，它把行为语言变成寄存器传送级语言，这时候你把它加到 Modelsim 中仿真叫后仿真，后仿真成功后，你还要在 quartus 中进行映射，布局布线，完后进行时序分析，生成时序网表，描述器件里门或者布线的延时，最后把延时网表和功能网表一起加到 Modelsim 中仿真叫门级仿真。

门级仿真和时序仿真的区别：门级仿真是 quartus 生成的网表文件.vo。门级则不考虑互联延迟，二只考虑了器件的延迟。时序仿真是选择具体器件并布局布线后进行的包含定时关系的仿真，主要验证是否满足时间约束关系、延时、最大工作频率和消耗的资源等。时序仿真是需添加时延文件.sdo。

理解方法四

从广义上讲，仿真验证包括功能与时序仿真和电路验证。仿真是指使用设计软件包对已实现的设计进行完整测试，模拟实际物理环境下的工作情况。从仿真的层次上划分，主要分为：

前仿真，也称为功能仿真或行为级仿真。是指仅对逻辑功能进行测试模拟，以了解其实现的功能是否满足原设计的要求，仿真过程没有加入时序信息，不涉及具体器件的硬件特性，如延时特性；

后仿真，也称为布局布线后仿真或时序仿真。是指提取有关的器件延迟、连线延时等时序参数，并在此基础上进行的仿真，它是非常接近真实器件运行情况的仿真。不同的工具和厂商还有一些其他的仿真过程，但大致属于这两类。

针对 FPGA 设计的流程，有 3 个阶段可以进行仿真：

第一阶段是寄存器传输级（RTL）仿真，此级仿真是对设计的语法和基本功能进行验证（不含时序信息）；

第二阶段是针对特定的 FPGA 厂家技术的仿真，此级仿真是在综合后、实现前而进行的功能级仿真，功能级仿真一般验证综合后是否可以得到设计者所需要的正确功能；

第三阶段是门级仿真，此级仿真是针对实现后的门级时序进行仿真，门级仿真体现了由于布局布线而产生的实际延时。

- **FPGA 学习交流：个人 QQ2793725733；**
- **Intel PSG FPGA 工程师培养基地：**
- **垂询：支老师（电话/微信：15529580559）；**